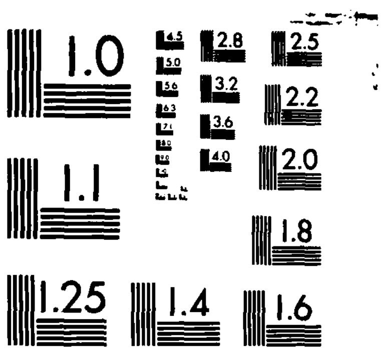
AD-A157 526 A NUMERICAL ANALYSIS OF A QUEUE WITH NETWORK ACCESS FLOW CONTROL(U) PRINCETON UNIV NJ DEPT OF ELECTRICAL ENGINEERING AND COMPUTER.. D W BROWNING ET AL. 1985 1/1

UNCLASSIFIED ARO-18985.19-EL DAAG29-82-K-0095 F/G 12/2 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A157 526

DTIC FILE COPY

(2)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ARO 18985.19-EL

2. GOVT ACCESSION NO.: N/A

3. RECIPIENT'S CATALOG NUMBER: N/A

4. TITLE (and Subtitle): A Numerical Analysis of a Queue with Network Access Flow Control

5. TYPE OF REPORT & PERIOD COVERED: Reprint

6. PERFORMING ORG. REPORT NUMBER:

AUTHOR(s): Douglas W. Browning and John B. Thomas

8. CONTRACT OR GRANT NUMBER(s): DAAG29-82-K-0095

PERFORMING ORGANIZATION NAME AND ADDRESS: Princeton Univ

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:

CONTROLLING OFFICE NAME AND ADDRESS: U. S. Army Research Office, Post Office Box 12211, Research Triangle Park, NC 27709

12. REPORT DATE:

13. NUMBER OF PAGES:

MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

NA

18. SUPPLEMENTARY NOTES

The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Numerical Analysis
Queueing Theory
Network Access Flow Control
Packet-Switched Networks

DTIC ELECTE AUG 0 7 1985 S D E

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ARO 18985.19-EL

PROCEEDINGS OF THE NINETEENTH ANNUAL CONFERENCE ON INFORMATION SCIENCES AND SYSTEMS

PAPERS PRESENTED

March 27, 28 and 29, 1985

Accession For
NTIS GRA&I ☒
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

PROGRAM DIRECTORS

G.G.L. Meyer
M.J. O'Donnell

Department of Electrical Engineering and Computer Science
The Johns Hopkins University
Baltimore, Maryland 21218

85 7 25 153

A NUMERICAL ANALYSIS OF A QUEUE WITH NETWORK ACCESS FLOW CONTROL

Douglas W. Browning and John B. Thomas

Department of Electrical Engineering and Computer Science
Princeton University

ABSTRACT

A network access flow control method is considered where messages are sent to the nodes which are major users of a congested queue. The model is based on a Markovian queue embedded in the network, and analysis takes place on a sequence of fixed-length time intervals. Changes of state reflecting the control and resulting input rates occur at the ends of the intervals, and the evolution of state probabilities over the course of an interval is analyzed numerically using Euler's method with Richardson Extrapolation [5]. The results provide insight into both transient and steady-state behaviors.

I. Introduction

There are several tools available in solving the problem of congestion in packet-switched networks. One of these is network access flow control, where packets are throttled at the input to a network in response to measurements of network congestion [2]. Network access flow control attempts to shift congestion to the boundaries of the network [1]. By only blocking packets that have not yet been admitted to the network, packets that have entered the network may be sent to their destinations without any major delays.

Network access flow control consists of a combination of monitoring network resources, exchanging information between nodes, and imposing restraints on packets entering the network. Measuring the state of resources and blocking entering packets can be easily performed by local switching processors. Consequently the main design problem is to determine the type and amount of information to be passed between the nodes. If too much information is exchanged, the network will be overloaded with flow control information. If too little information is exchanged, the flow control will not be able to respond properly to many congestion situations, so that additional mechanisms must be implemented to ensure proper network behavior. The desirable design is that where the nodes exchange just enough information to handle well all reasonable congestion situations.

An important aspect of network access flow control is whether it performs effectively under most of the conditions that arise in the operation of a network. A simple solution to this problem may not exist, since the most easily implemented schemes are either ineffectual in at least some situations or have other problems related to overhead or reliability. For example, when an Input Buffer Limit [2] is used, the implementation is simple and there is no overhead communication between nodes, but it is quite easy to develop scenarios where congestion is not corrected and nodes where there is heavy traffic from other sources are denied access to the network.

This paper considers a scheme where information is passed as the need arises from a point of congestion to the major sources using the congested resource. At the cost of some additional overhead and complexity, the scheme is designed to respond effectively to almost all congestion situations. A similar approach is used in the mechanism developed for the CYCLADES network [2,4].

The design objective used here is to limit the throughput over each channel to 0.70 times the channel capacity, so that the delay through each queue will be no more than roughly 3 times the minimum possible delay.

II. Flow Control Method

The control consists of interaction between a node with a congested queue and the sources which send packets through that queue. Each node maintains a table of recently sent control information and a record of recent traffic for each of its queues, from which the major users are determined.

Whenever a packet enters a queue, the switching processor measures the length of the queue. If the length exceeds a threshold, the queue is considered to be congested. Then a *control message* is generated and sent towards those sources which have not been sent a control message for some short period of time and which have recently been major users of the queue.

In the absence of control messages, each source limits the rate of traffic to each destination to 0.70 times the slowest service rate that will be encountered along its route. When a control message is received, traffic towards the congested queue is stopped for a very brief period of time. Then, by requiring successively shorter delays between packets, the rate is allowed to increase gradually (for example, linearly in time) until the load limit of 0.70 is again reached.

This method is intended to operate effectively under all steady-state conditions, and to respond as quickly as possible to changes in arrival rates. By establishing messages between a congested queue and a packet source as the need arises, all parts of the network may communicate, yet the overhead is maintained at a low level.

It should be noted that each control message is a self-contained unit which has an effect for only a finite period of time, so there are no problems with removing control, and the control message need not be acknowledged. As long as the network is fairly reliable, an occasional lost or garbled control message will not have any major adverse effects.

III. Model

This flow control method is modeled as a single queue embedded in a network with arriving packets entering the queue after a delay proportional to the distance from the source to the queue, as shown in Figure 1.

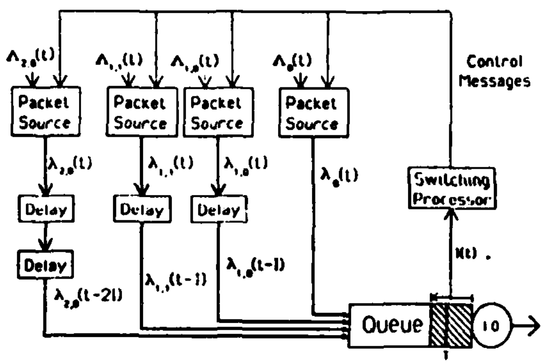


Figure 1. Model of a Queue with Flow Control. T is the threshold; $l(t)$ is the length of the queue; $\Lambda_{k,i}(t)$ is the offered traffic rate of source i at distance k; $\lambda_{k,i}(t)$ is the accepted traffic rate; and delays are of fixed length I.

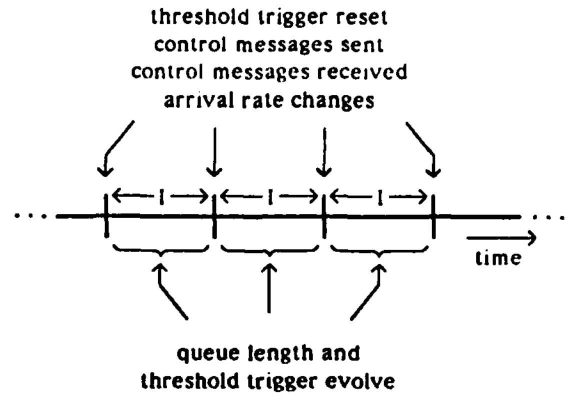


Figure 2. Division of Time into Intervals.

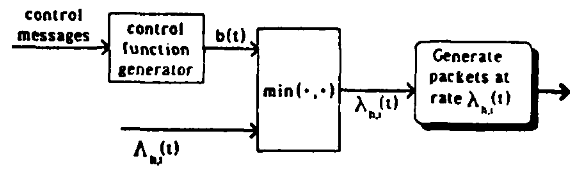


Figure 3. Model of Packet Source. $\Lambda_{k,i}(t)$ is the offered traffic rate of source i at distance k; $\lambda_{k,i}(t)$ is the accepted traffic rate. The control function generator limits the traffic according to the function $b(t)$ described in the text.

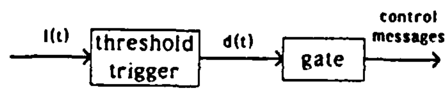


Figure 4. Model of the Switching Processor. $l(t)$ is the length of the queue; $d(t)$ is the threshold trigger. The threshold trigger sets to 1 when $l(t) \geq T$, and resets to 0 at the end of each interval where $l(t) < T$. The gate sends a control message at the end of each interval where the last control message was sent more than S intervals before and $d(t) = 1$.

The outgoing channel and its queue are modeled as a Markovian queue with a buffer limit, the control message generation by the switching processor is modeled by a threshold trigger and a gate, the behavior of each source node is modeled by a Poisson packet source where the renewal rate is determined by an offered traffic load and the throttling resulting from control messages, and the times for all other queueing and transmissions in the network are modeled as delays of fixed length.

Delays. The delays model the time necessary for generating a control message and sending it through the network to a source plus the time necessary after a source begins changing the rate of packets entering the system for the change to take effect at the input to the queue. The time required for this to happen is assumed to be I times the distance to the congested queue, where I is a fixed period of time.

The state of the model evolves over a series of intervals of duration I, with the queue receiving and serving packets within those intervals. Control messages are generated, sent, and take effect at the queue at the ends of the intervals. See Figure 2.

Packet sources. If a source receives a control message, it removes any control previously in effect and places an upper bound $b(t)$ on the rate at which traffic will pass through the congested queue. The function $b(t)$ is 0.0 for one time interval, and then increases by a fixed increment at the end of each following time interval until the load limit of 0.70 is reached.

The function $b(t)$ takes on the value of one of N *control levels* a_j, which are linearly interpolated between 0.7 and 0.0.

$$a_j = 0.7\left(1-\frac{j}{N-1}\right) \quad , j = 0, \ldots, N-1$$

When a control message is received at a source, $b(t)$ is set to a_{N-1} (0.0). The control is relaxed to the next larger level in each following interval until another control message is received, resetting it to a_{N-1}. If the control reaches a_0 (0.7), it remains constant until another control message is received. Thus, $b(t)$ can be given by

$$b(t) = \min\left[0.7\ ,\ 0.7\left(1-\frac{k}{N-1}\right)\right]$$

where

$$k = \min\{\, j :\ \textit{a message arrived in } [t-(j+1)I, t-jI)\}$$

Each source generates a Poisson process of packets at rate $\lambda_{h,i}$, where h,i indicates the ith source at the distance of h hops from the queue. This rate is determined by the minimum of the offered traffic rate $\Lambda_{h,i}$ and the upper bound imposed by the control function $b(t)$, as shown in Figure 3; thus

$$\lambda_{h,i}(t) = \min\,[\Lambda_{h,i}(t), b(t)]$$

Switching processor. The switching processor is modeled as a trigger, which detects congestion, and a gate which ensures a minimum spacing between control messages, as in Figure 4. If the length of the queue $l(t)$ exceeds a threshold T, a trigger $d(t)$ is activated. Then at the end of a time interval, if enough time has elapsed since the last control message, a new control message is sent to all of the major users of the queue.

When a control message is sent, another may not be sent until S intervals have passed. Requiring a spacing S between control messages serves both to reduce the number of control messages and to allow the system enough time to respond to previous messages before sending additional ones.

Queue. The server has an exponential service time distribution with an expected service time normalized to one second.

The arrival rate $A(t)$ is scaled in terms of the traffic load on the queue and is given by

$$A(t) = \sum_{h,i} \lambda_{h,i}(t-hI)$$

The buffer is able to hold at most B packets, including the packet receiving service. When the buffer is full, further arrivals are discarded. The length of the queue is measured continuously by the switching processor.

State Description

The state of the model can be described in terms of the set of variables (l, d, C), where

- l is the length of the queue in packets. $0 \leq l \leq B$, where B is the buffer size.
- d is the threshold trigger. $d=1$ if the threshold has been exceeded during the course of the present interval, otherwise $d=0$.
- C is the control state. It is a vector $(c_0, c_1, c_2, \ldots,)$ where c_h is the index of the control level s_{c_h}, which determines the maximum rate of traffic entering the queue from each source at the distance h. When a control message is generated, c_0 is set to $N-1$. At the end of each following interval, it is decremented by 1 until it reaches 0 or until another control message is generated. For $h > 0$, c_h is simply c_0 delayed by h intervals.

As can be seen from Figure 6, the number of possible states of C is fairly limited. The number of possible states of the pair (l, d) is $B+1+T$, so that for reasonable values of the system parameters the state space will be limited to a few hundred states. Because of this, a numerical analysis can be performed easily.

IV. Analysis

The numerical method produces a transient analysis of the state probabilities at the ends of time intervals. Within each interval, the control state remains constant while only the queue length and threshold trigger state probabilities evolve. At the end of each interval, the control state C changes instantaneously according to the previous control state and the threshold trigger d, as shown in Figure 6.

Interval dynamics

The evolution of the queue state probabilities over each interval is analyzed from the system dynamics using Euler's method with Richardson extrapolation, as is described in most books on Numerical Analysis [5, 6]. For an arrival rate $A(t)$, the state transition rates of the queue and threshold trigger during the course of an interval are shown in Figure 5.

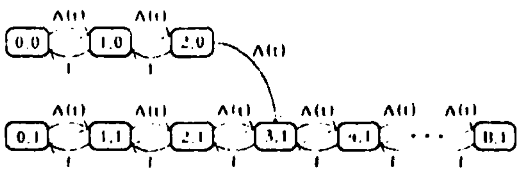


Figure 5. State Transition Rates within Interval. The states are given as (l, d). For this diagram $T=3$.

For a constant arrival rate A, the dynamics of the queue length probability $p_{l,d}$ for queue length l and threshold trigger d for $1 < T < B$ are given by

$$\frac{\partial}{\partial t} p_{0,d} = -p_{0,d} A + p_{1,d}$$

$$\frac{\partial}{\partial t} p_{B,1} = -p_{B,1} + p_{B-1,1} A$$

$$\frac{\partial}{\partial t} p_{T,1} = -p_{T,1}[A+1] + p_{T-1,0} A + p_{T-1,1} A + p_{T+1,1}$$

$$\frac{\partial}{\partial t} p_{T-1,0} = -p_{T-1,0}[A+1] + p_{T-2,0} A$$

$$\frac{\partial}{\partial t} p_{l,d} = -p_{l,d}[A+1] + p_{l-1,d} A + p_{l+1,d}$$

where the last equation holds for all possible states where none of the previous equations apply.

For a small time interval Δt, these equations may be approximated by the first order difference equations

$$p_{0,d}(t+\Delta t) \approx p_{0,d}(t)[1-A\Delta t] + p_{1,d}(t)\Delta t$$

$$p_{B,1}(t+\Delta t) \approx p_{B,1}(t)[1-\Delta t] + p_{B-1,1}(t) A \Delta t$$

$$p_{T,1}(t+\Delta t) \approx p_{T,1}[1-(A+1)\Delta t] + p_{T-1,0}(t) A \Delta t + p_{T-1,1}(t) A \Delta t + p_{T+1,1}(t)\Delta t$$

$$p_{T-1,0}(t+\Delta t) \approx p_{T-1,0}(t)[1-(A+1)\Delta t] + p_{T-2,0} A \Delta t$$

$$p_{l,d}(t+\Delta t) \approx p_{l,d}(t)[1-(A+1)\Delta t] + p_{l-1,d}(t) A \Delta t + p_{l+1,d}(t)\Delta t$$

where again the last equation holds when none of the previous apply.

For the evolution of the queue length probabilities, the numerical method uses an iterative application of these difference equations.

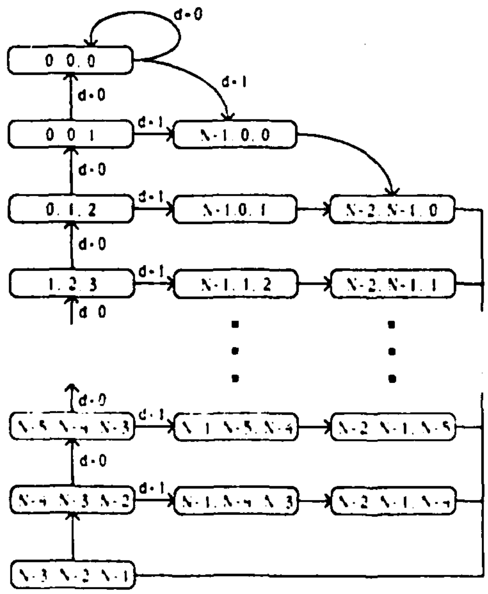


Figure 6. State Transitions at Ends of Intervals. All states change deterministically depending on the state and the threshold trigger d. For this diagram $S=4$, and the maximum distance from the queue is 2.

As is well known in numerical analysis [5,6], if the difference approximation is used with the time interval divided into short intervals of equal length Δt, the error $e_{\Delta t}$ between the actual value and the approximated value will become proportional to Δt as Δt becomes small.

$$e_{\Delta t} = \kappa \times \Delta t + o(\Delta t)$$

where κ is a constant and $f(\Delta t) = o(\Delta t)$ indicates

$$\lim_{\Delta t \to 0} \frac{f(\Delta t)}{\Delta t} = 0$$

If the interval is successively divided into sub-intervals of length Δt, $\frac{\Delta t}{2}$, and $\frac{\Delta t}{4}$, the errors should have the relationship

$$e_{\frac{\Delta t}{4}} \approx \frac{1}{2} e_{\frac{\Delta t}{2}} \approx \frac{1}{4} e_{\Delta t} \approx \frac{1}{4} \kappa \times \Delta t$$

Thus, if the number of divisions is doubled until the $o(\Delta t)$ error becomes small enough, the $\kappa \times \Delta t$ error can be removed by extrapolation.

Endpoint transitions

At the end of each interval, each state is mapped into a new state depending on whether or not a control message is generated for that state, as in Figure 6. The new state is determined as follows:

If $d=1$ and $c_0 \leq N-S$, then congestion has been detected and sufficient time has elapsed since the last control message, so that a control message is generated. Each c_h for $h>0$ is replaced by c_{h-1}, and c_0 is set to $N-1$.

If $d=0$ or $c_0 > N-S$, then a control message is not generated. Each c_h for $h>0$ is replaced by c_{h-1}, and c_0 is decremented by 1, unless it is 0, in which case it remains constant.

If $l<T$, then d is set to 0, otherwise it is set to 1.

V. Parameters

System Parameters

For the numerical analysis, the buffer size B was set at 9 packets, and the time interval I was set at 3.5 seconds. The latter was chosen as a rough estimate of the delay through a well controlled queue under congestion conditions, plus a small time for any other overhead in control message transmission and processing. The control messages are assumed to have priority over other packets, so that the delay in reaching the source node is relatively small.

Design Parameters

Once the system parameters are fixed, there are three parameters which may be set by a flow control designer to achieve the best performance:

The threshold T, which is the level of buffer occupancy which activates the threshold trigger.

The spacing S, which sets a minimum on the number of intervals between control messages.

The number of control levels N, which determines the amount of traffic held back at a source as the result of a single control message.

The interaction between these three parameters will be explored in the Results.

Output Parameters

Three output parameters which can be calculated from the analysis are the transient *expected throughput*, which is simply the probability of one or more packets being in the queue

$$Pr[l>0]$$

the *expected delay*, which is the expected time that a packet entering the queue will wait before service is completed, given by

$$I + \frac{\sum_{l=0}^{B-1} (l \times Pr[l])}{\sum_{l=0}^{B-1} Pr[l]}$$

and the *expected overhead*, which is the probability that a control message will be sent at the end of an interval

$$P[d=1]$$

VI. Results

Input

The model was analyzed with the initial state where the queue is initially empty with no control in effect (that is, $l=0$, $d=0$, $C=(0,0,0,\ldots)$). At time 0.0, the queue suddenly receives a maximum input from all active sources. Since the situation of primary interest is that where the network is congested, all sources are assumed to provide the maximum possible input to the network, except in the section on non-congestion behavior. This provides a view of the transient response to an abrupt change in input and also allows a simple initialization.

Configurations of sources at distances of up to three hops from the host were considered. In subsequent diagrams, the arrangement of sources will be denoted by a sequence of digits

$$X_0 X_1 X_2 X_3$$

where X_h indicates the number of sources at a distance of h. For example, the configuration

1201

indicates that the host node is switching packets to the queue, and packets are also being sent to it by two sources at a distance of 1 and one source at a distance of 3.

Both transient and steady state aspects of the behavior of this model were examined.

Transient Behavior

Spacing. The results of varying the minimum spacing S between control messages are shown in Figure 7. Early in the analysis, control messages were permitted to be transmitted in adjacent time intervals ($S=1$). The result was that when the queue became congested, a control message was sent, and before the control was able to take full effect, a second and sometimes a third control message would also be sent. This would cause over-compensation, so that the expected input would drop to a very low level for a short time and then begin rapidly increasing again, causing large oscillations in the transient expected throughput.

By requiring a delay between control messages, this problem was corrected with the bonus of reducing overhead. When the time delay between messages becomes too large, however, it was found that the effect of the control begins to disappear before the next control message can be sent, so that oscillations again occur ($S=6$). The spacing also has a major effect on the steady-state throughput and delay, so that they must be readjusted by other means.

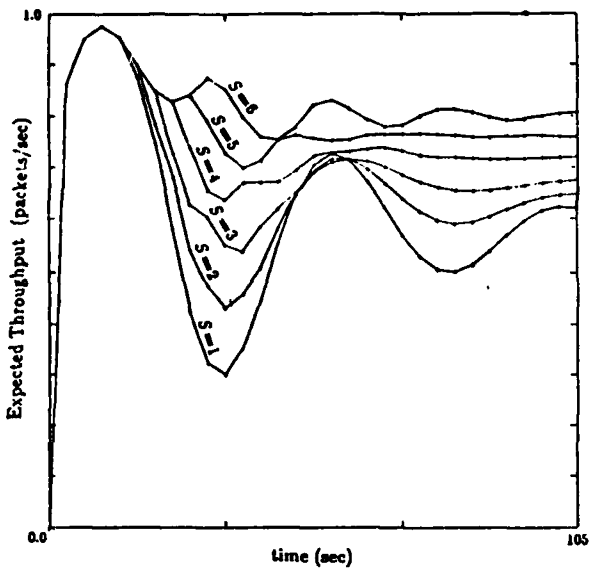


Figure 7. Transient Expected Throughput with Various Values of S. These results are for the configuration 0020 with $N=9$ and $T=3$.

Detection threshold. The transient expected throughput is considered in Figure 8 with several different values of T. Here the distinctive change in behavior is that as the threshold becomes smaller, the system is responds more quickly to a surge in input. There are also more control messages sent when they aren't needed, so that the expected overhead increases as the threshold decreases. Again there is a major effect on the steady state throughput and delay, so that they must be readjusted by other means.

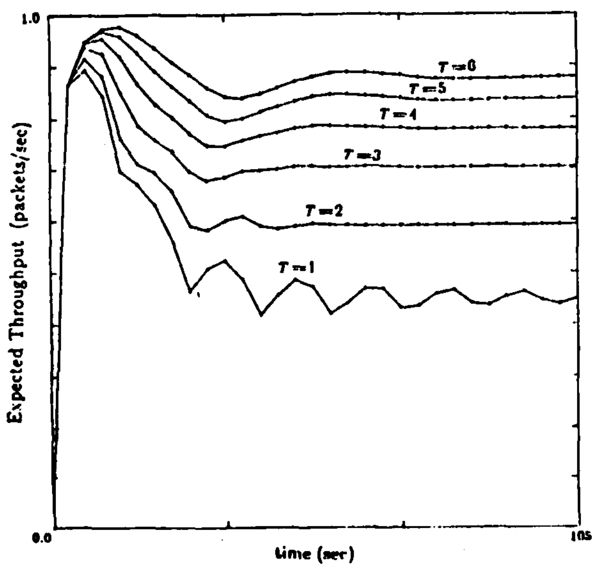


Figure 8. Transient Expected Throughput with Various Values of T. These results are for the configuration 1110 with $N=9$, $S=4$.

Control levels. Analyses, such as that shown in Figure 9, indicate that by varying N, the steady-state throughput can be adjusted while only slightly affecting the quickness of the response and the amount of oscillation of the transient expected throughput. Thus if S and T are chosen to produce a desired shape for the transient expected throughput, the value of N which produces a steady-state expected throughput near 0.70 can then be chosen.

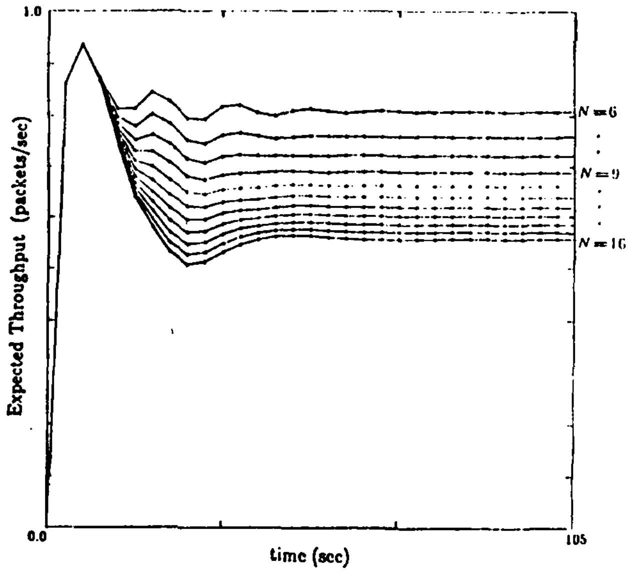


Figure 9. Transient Expected Throughput with various values of N. These results are for the configuration 1200 with $S=4$, $T=3$.

Steady State Behavior

The choice of T and S. For various configurations of sources, a variety of combinations of N, T, and S which result in a throughput near 0.7 were considered. Some typical results are shown in Table 1. The transient performance for these results is shown in Figure 10. While it is possible to examine every configuration of sources in order to find the "best" choice of N, T, and S for each situation, a more practical choice is to find a set of values which is easily determined and which works well under a variety of configurations. The values $S=4$ and $T=3$ resulted in a fairly fast response and modest oscillations in the transient expected throughput for all configurations examined. The following analyses were made with S and T at these values.

Table 1

Overhead for Various Combinations of Parameters

S	T	N	expected overhead
3	3	8	.21
	4	12	.15
	5	17	.11
4	2	7	.21
	3	9	.17
	4	13	.13
	5	18	.10

Table 1. Steady State Overhead for Combinations of Parameters with with a Steady State throughput near 0.70. N is chosen so that the steady state throughput is as near as possible to 0.70. The configuration is 1110.

Non-congestion behavior. The flow control scheme worked effectively in all steady state situations examined.

Figure 11 shows that for the configuration examined, the throughput is limited to a value near 0.7, and the delay is limited to a value near 3.0, so that the design objective is met. Also, it is shown that the relationship of expected delay to throughput in non-congestion conditions is similar to that without the flow control.

The throughput as a function of offered load in Figure 12 demonstrates the limiting of throughput to 0.7 times the channel capacity. The resulting throughput versus accepted traffic in Figure 13 demonstrates the effective limiting of traffic entering the system, so that the network operates within a desireable region.

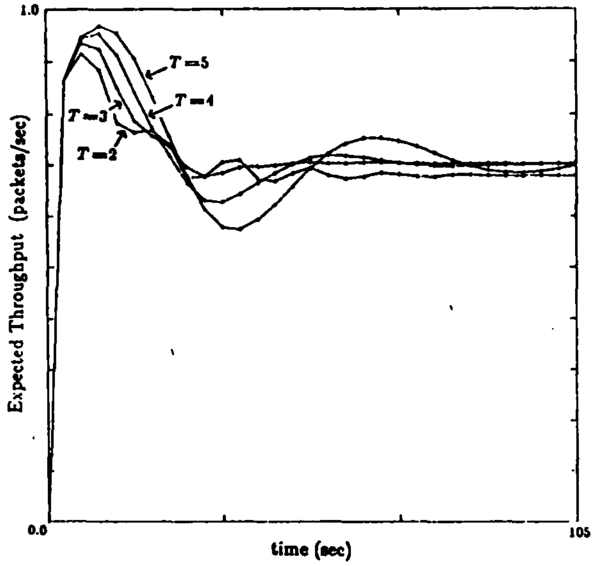


Figure 10. Transient Expected Throughput with a Steady State Throughput near 0.70. The plotted values are the transient expected throughputs with the parameters of Table 1 where $S=4$.

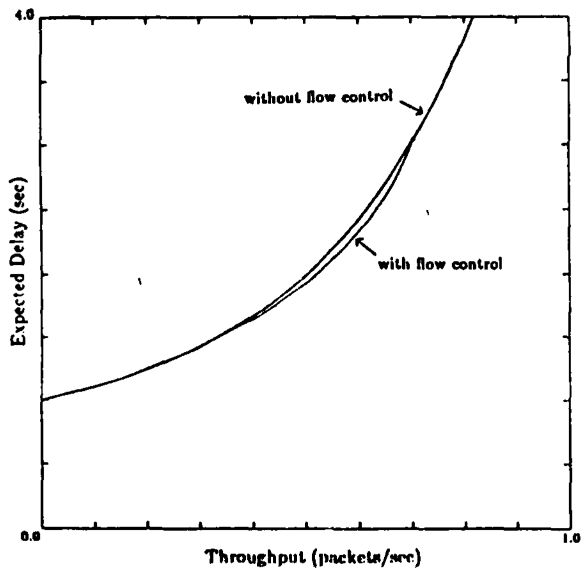


Figure 11. Steady State Expected Delay versus Throughput. These results are for the configuration 1110 with $N=9$, $S=4$, and $T=3$. Under non-congestion conditions, each source is offered traffic at the same rate.

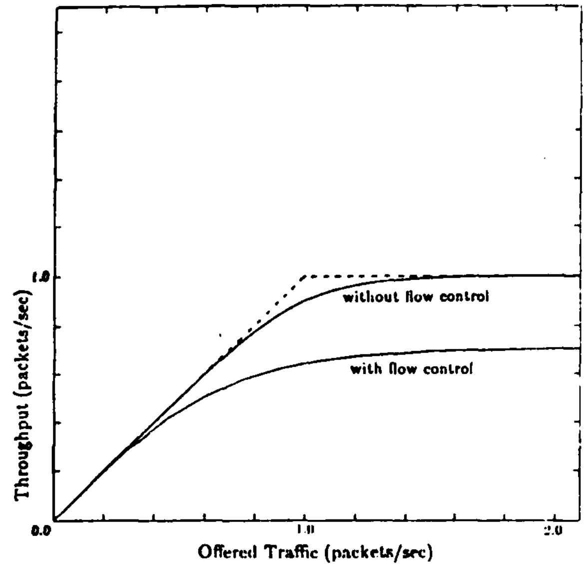


Figure 12. Steady State Throughput versus Offered Traffic. These results are under the same conditions as Figure 11. The dashed lines indicate theoretical limits on the throughput.

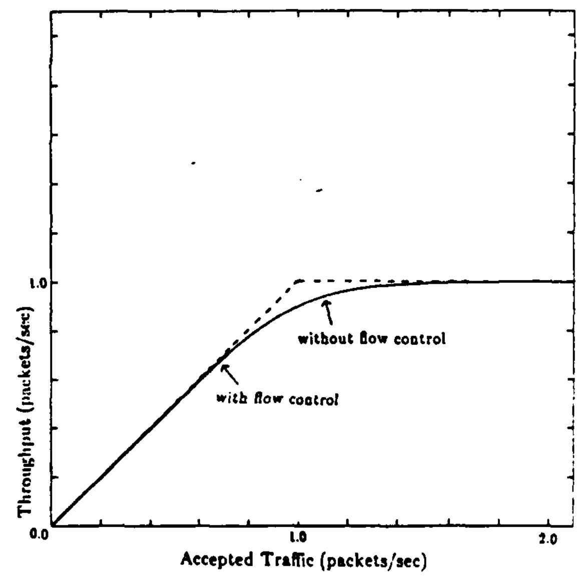


Figure 13. Steady State Throughput versus Accepted Traffic. These results are under the same conditions as Figure 11. The dashed lines indicate theoretical limits on the throughput.

Congestion behavior With $S=4$ and $T=3$, a value of N should be chosen so that the steady state throughput and delay are near the design values of 0.7 and 3.0. If a constant value of N is chosen, as was done in the results of Figure 14, the steady state results are closely clumped into groups with the same number of sources. A simple relation which produces steady state values near the design values is that where N is proportional to the number of sources sending packets towards the queue. This number is easily determined by the switching processor of the queue and is easily parameterized for inclusion in the control message. Figure 15 shows that this is quite effective. The exception is the case where there is only one active source using the queue. In that case, the best action is actually no control at all, since the throughput would then be no more than 0.7.

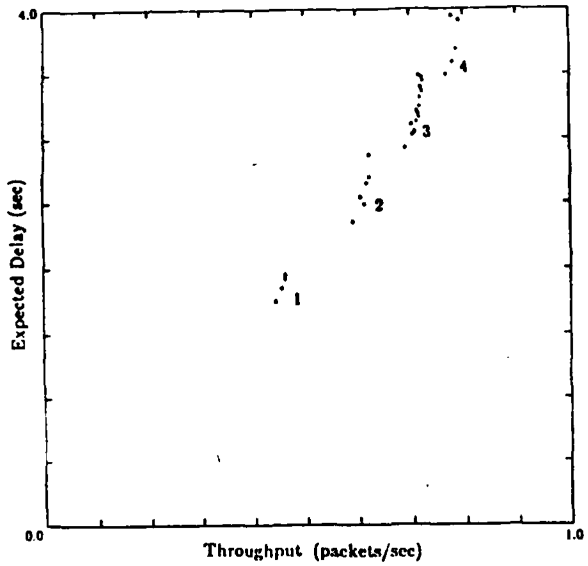


Figure 14. Steady State Expected Delay versus Throughput under Congestion Conditions with a Constant value of N. Each point is the steady state operating point for a particular configuration of sources sending packets towards the queue. Each cluster of consists of results for configurations with the indicated number of sources. Here $N=9$.

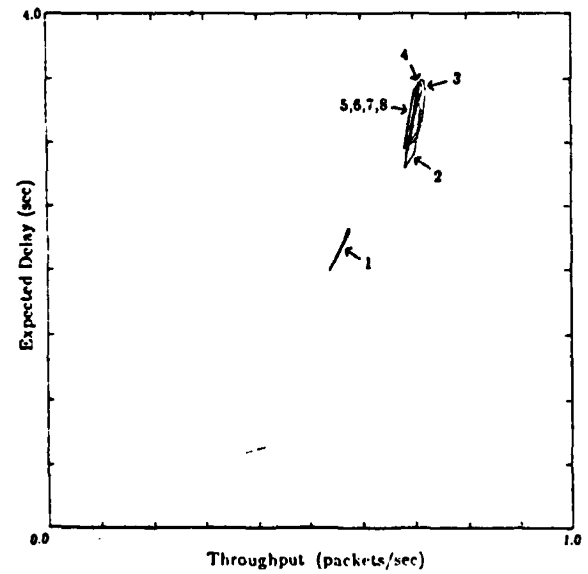


Figure 15. Steady State Expected Delay versus Throughput under Congestion Conditions with N Proportional to the Number of Sources. Each polygon is the smallest convex polygon enclosing the steady-state operating points of the configurations examined with the indicated number of sources. Here N is three times the number of sources.

VII. Conclusions

The results show that the described method is quite effective in controlling congestion that arises from steady-state input, and that the control can be implemented using simply chosen design parameters. The transient results demonstrated the effectiveness of the control method in response to a surge in traffic, and established the following interaction of design parameters:

The threshold at which congestion is detected is related to the quickness of the response of the control.

The minimum spacing between control messages is related to the amount of oscillation of the transient expected throughput.

The amount of traffic held back as a result of a single control message affects the steady state throughput while affecting the shape of the transient behavior only slightly.

Several questions remain unresolved. The response of the system to a drop in traffic from one or several users is still undetermined. Also, the response to extremely bursty traffic from several different sources has not been investigated. Simulations to test the accuracy of the model are also necessary.

Acknowledgements

This work was supported in part by the U. S. Army under Grant DAAG29-82-K-0095 and in part by the National Science Foundation under Grant ECS83-17777.

Much of the groundwork upon which this is based was performed at Oki Electric Company, Tokyo, Japan supported by an American Electronics Association Japan Research Fellowship.

References

[1] Jeremiah Hayes, *Modeling and Analysis of Computer Communication Networks*, Plenum Press, 1984.

[2] Mario Gerla and Leonard Kleinrock, "Flow control protocols," in Paul E. Green, Jr., ed., *Computer Network Architectures and Protocols*, Plenum Press, 1982. pp. 361-412.

[3] L. Pouzin, "Methods, tools, and observations on flow control in packet-switched data networks," *IEEE Transactions on Communication*, Vol. COM-29, 4 (April 1981), pp. 413-426.

[4] Majithia, *et. al.*, "Experiments in congestion control techniques," in *Proceedings of the International Symposium on Flow Control in Computer Networks*, Versailles, France, Feb. 1979.

[5] Germund Dahlquist and Ake Bjork *Numerical Methods* Prentice-Hall, 1974.

[6] Eugene Isaacson and H. B. Keller, *Analysis of Numerical Methods*, Wiley, 1966.

END

FILMED

9-85

DTIC